

CLAIMS

We claim:

1. A method for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said method comprising the steps of:

determining which software libraries are potentially needed during execution of the application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring anytime before any library of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module;

executing the loaded library implementation module for the library routine being executed; and

unloading from memory the library implementation module after the execution of the library routine is completed.

2. The method as defined in Claim 1 wherein said loading of a library loader for each potentially needed library occurs either just prior to or contemporaneous with the execution of the application or other software module.
3. The method as defined in Claim 1 wherein said loading of a library loader for each potentially needed library occurs after execution of the application or other software module begins

but before any library of the potentially needed library is executed by the application or other software module.

4. A method for dynamically loading a software library into memory in a computer, said software library capable of being used by an application or other software module, said method comprising the steps of:

determining which libraries are potentially needed during execution of the application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring anytime before any library of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module, said loading comprising the steps of:

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module; and

if the library implementation module is in memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

scheduling the unloading of the library implementation module.

5. The method as defined in Claim 4 wherein said loading of a library loader for each potentially needed library occurs either just prior to or contemporaneous with the execution of the application or other software module.
6. The method as defined in Claim 4 wherein said loading of a library loader for each potentially needed library occurs after execution of the application or other software module begins but before any library of the potentially needed library is executed by the application or other software module.
7. The method as defined in Claim 4 wherein said step of scheduling the unloading of the library implementation module includes setting an associated criteria and unloading the library implementation module if after said associated criteria has been met, said library implementation module is not in use.
8. The method as defined in Claim 7 wherein said associated criteria is a time delay.
9. The method as defined in Claim 8 wherein said time delay is fixed.
10. The method as defined in Claim 8 wherein said time delay is variable.
11. The method as defined in Claim 7 wherein said associated criteria is based on system resources.
12. The method as defined in Claim 4 wherein said step of scheduling the unloading of the library implementation module comprises the steps of:

setting an associated criteria;

unsetting an associated criteria if said scheduled unloading is cancelled; and

resetting the associated criteria if the library implementation module is again scheduled for unloading.

13. A method for dynamically loading a software library into memory in a computer, said software library capable of being used by a plurality of applications or other software modules, said method comprising the steps of:

loading a software library into memory just prior to or contemporaneous with execution of a library routine in said software library; and

unloading said software library from memory after completion of execution of said library routine.

14. A method for dynamically loading a software library into memory in a computer, said software library capable of being used by an application or other software module, said method comprising the steps of:

loading a software library into memory just prior to or contemporaneous with execution of a library routine in said software library; and

scheduling said software library from memory after completion of execution of said library routine.

15. The method as defined in Claim 14 wherein said step of scheduling the unloading of the library implementation module includes setting an associated criteria and unloading the library

implementation module if after said associated criteria has been met, said library implementation module is not in use.

16. The method as defined in Claim 15 wherein said associated criteria is a time delay.
17. The method as defined in Claim 16 wherein said time delay is fixed.
18. The method as defined in Claim 16 wherein said time delay is variable.
19. The method as defined in Claim 15 wherein said associated criteria is based on system resources.
20. The method as defined in Claim 14 wherein said step of scheduling the unloading of the library implementation module comprises the steps of:

setting an associated criteria;

unsetting an associated criteria if said scheduled unloading is cancelled; and

resetting the associated criteria if the library implementation module is again scheduled for unloading.

21. A library structure for a software library, said library structure stored in a computer-readable medium, said library structure comprising:

a library loader specifying entry points corresponding to entry points in the software library; and

a library implementation module containing computer code to implement the software library,

wherein said library loader contains code for loading and unloading said library implementation module from memory.

22. The library structure as defined in Claim 21, wherein said library loader is loaded into memory at or near the beginning of execution of an application program or other software module referencing said software library.
23. The library structure as defined in Claim 21, wherein said library loader is loaded into memory such that said library loader is in memory prior to a call of said software library by an application program or other software module.
24. A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software for performing the following steps:

determining which libraries are potentially needed during execution of the application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring anytime before any library of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module;

executing the loaded library implementation module for the library routine being executed; and

unloading from memory the library implementation module after the execution of the library routine is completed.

25. The computer system as defined in Claim 24 wherein said loading of a library loader for each potentially needed library occurs either just prior to or contemporaneous with the execution of the application or other software module.
26. The computer system as defined in Claim 24 wherein said loading of a library loader for each potentially needed library occurs after execution of the application or other software module begins but before any library of the potentially needed library is executed by the application or other software module.
27. A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software for performing the following steps:

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module, said loading comprising the steps of:

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module; and

if the library implementation module is in memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

scheduling the unloading of the library implementation module.



28. The computer system as defined in Claim 27 wherein said loading of a library loader for each potentially needed library occurs either just prior to or contemporaneous with the execution of the application or other software module.
29. The computer system as defined in Claim 27 wherein said loading of a library loader for each potentially needed library occurs after execution of the application or other software module begins but before any library of the potentially needed library is executed by the application or other software module.
30. The computer system as defined in Claim 27 wherein said step of scheduling the unloading of the library implementation module includes setting an associated criteria and unloading the library implementation module if after said associated criteria has been met, said library implementation module is not in use.
31. The computer system as defined in Claim 30 wherein said associated criteria is a time delay.
32. The computer system as defined in Claim 31 wherein said time delay is fixed.
33. The computer system as defined in Claim 31 wherein said time delay is variable.
34. The computer system as defined in Claim 30 wherein said associated criteria is based on system resources.
35. The computer system as defined in Claim 27 wherein said step of scheduling the unloading of the library implementation module comprises the steps of:

setting an associated criteria;

unsetting an associated criteria if said scheduled unloading is cancelled; and

resetting the associated criteria if the library implementation module is again scheduled for unloading.

36. The computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software for performing the following steps:

loading a software library into memory just prior to or contemporaneous with execution of a library routine in said software library; and

unloading said software library from memory after completion of execution of said library routine.

37. The computer system of Claim 36 wherein said unloading of said software library comprises the steps of:

scheduling the unloading of the library implementation module;

setting an associated criteria;

checking whether said set associated criteria has been met; and

unloading the library implementation module if after said associated criteria has been met, said library implementation module is not in use.

38. A computer-readable medium having stored thereon instructions for causing a computer to perform the following steps:

determining one or more software libraries which are potentially needed during execution of an application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring anytime before any library of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module;

executing the loaded library implementation module for the library routine being executed; and

unloading from memory the library implementation module after the execution of the library routine is completed.

39. The computer-readable medium as defined in Claim 38 wherein said loading of a library loader for each potentially

needed library occurs either just prior to or contemporaneous with the execution of the application or other software module.

40. The computer-readable medium as defined in Claim 38 wherein said loading of a library loader for each potentially needed library occurs after execution of the application or other software module begins but before any library of the potentially needed library is executed by the application or other software module.

41. A computer-readable medium having stored thereon instructions for causing a computer to perform the following steps:

determining one or more software libraries which are potentially needed during execution of an application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring anytime before any library of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module, said loading comprising the steps of:

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module;

if the library implementation module is in memory,  
then canceling the scheduled unloading of the library  
implementation module;

executing the loaded library implementation module for  
the library routine being executed; and

scheduling the unloading of the library implementation  
module.

42. The computer-readable medium as defined in Claim 41  
wherein said loading of a library loader for each potentially  
needed library occurs either just prior to or contemporaneous  
with the execution of the application or other software module.
43. The computer-readable medium as defined in Claim 41  
wherein said loading of a library loader for each potentially  
needed library occurs after execution of the application or  
other software module begins but before any library of the  
potentially needed library is executed by the application or  
other software module.
44. The computer-readable medium as defined in Claim 41  
wherein said step of scheduling the unloading of the library  
implementation module includes setting an associated criteria  
and unloading the library implementation module if after said  
associated criteria has been met, said library implementation  
module is not in use.
45. The computer-readable medium as defined in Claim 44  
wherein said associated criteria is a time delay.
46. The computer-readable medium as defined in Claim 45  
wherein said time delay is fixed.

47. The computer-readable medium as defined in Claim 45 wherein said time delay is variable.
48. The computer-readable medium as defined in Claim 44 wherein said associated criteria is based on system resources.
49. The computer-readable medium as defined in Claim 41 wherein said step of scheduling the unloading of the library implementation module comprises the steps of:
- setting an associated criteria;
  - unsetting an associated criteria if said scheduled unloading is cancelled; and
  - resetting the associated criteria if the library implementation module is again scheduled for unloading.
50. A computer-readable medium having stored thereon instructions for causing a computer to perform the following steps:
- loading a software library into memory just prior to or contemporaneous with execution of a library routine in said software library; and
  - unloading said software library from memory after completion of execution of said library routine.
51. A method for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said method comprising the steps of:

determining which software libraries are potentially needed during execution of the application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring anytime before any library of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library containing a library routine, said loading occurring after the start of execution of the application or other software module and prior to when said library routine is to be executed by the application or other software module;

executing the loaded library implementation module for the library routine being executed; and

unloading from memory the library implementation module after the execution of the library routine is completed.